O que é o algoritmo K-means?

João Paulo de Carvalho Araújo - 202065564C
Lucas Tassi Facciolla - 201935025

Resumo

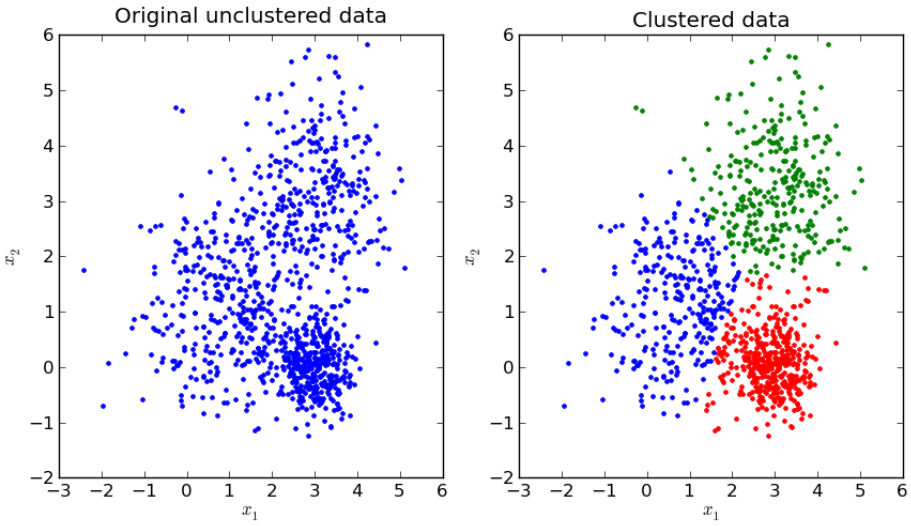
Neste trabalho serão abordados os algoritmos k-means, tratando inicialmente do aprendizado não-supervisionado e pelas técnicas de clusterização para dar base ao conteúdo.

Aprendizado não-supervisionados

Aprendizado não-supervisionado (ou aprendizado de máquina não-supervisionado) se utiliza de algoritmos para análise e agrupamento de dados não classificados. Estes algoritmos são capazes de encontrar tendências e agrupar dados sem intervenção externa, o que o torna a solução ideal para análise exploratória de dados, reconhecimento de imagem, segmentação de clientes, etc.

Clusters

Técnicas de clustering são amplamente utilizadas no âmbito de machine learning, data mining, image segmentation, exploração de dados, etc. Um cluster nada mais é do que um grupo de dados que possuem similaridade entre si, ou seja, coerência, podendo existir vários clusters por conjunto de dados. Abaixo um exemplo de cluster sobre um dataset aleatório:



Retirado em K-Means Data Clustering

Existem diversos algoritmos de clusterização. Estes preenchem o grupo de métodos de aprendizado não-supervisionados, isto é, não possuem uma classe/label associada a cada exemplo. A seguir são apresentados 3 tipos diferentes de abordagens bastante populares

utilizadas para realizar agrupamento de dados: Clusterização baseada em centróide, clusterização hierárquica e clusterização baseada em densidade.

- **Métricas de similaridade**

Para medir a proximidade entre os pontos de dados, são necessárias métricas capazes de avaliar o quão semelhantes diferentes pontos são entre si. Como na maioria dos casos utiliza-se distância euclidiana, apenas 2 serão apresentadas:

- Distância euclidiana: métrica mais utilizada, mede a distância entre dois pontos de dados. Toma como cálculo a norma entre a soma do quadrado entre os pontos de dados e o centro do cluster.

$$\text{Euclidean distance:}$$
$$d = \sqrt{\sum (X - Y)^2}$$

Retirado em A Similarity Measure for Clustering and Its Applications

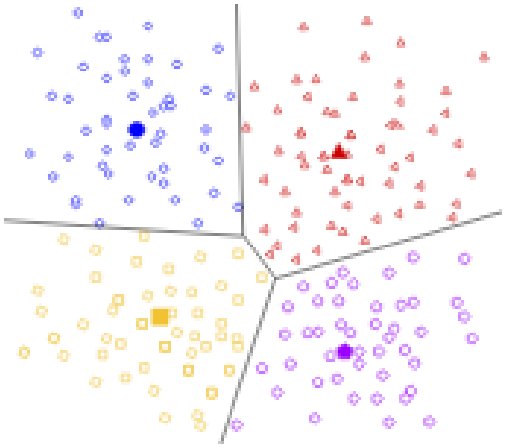
- Similaridade de cossenos: diferente da distância euclidiana, a métrica de similaridade entre cossenos leva em conta a distância angular entre os 2 pontos de dados. Essa métrica é mais utilizada para documentos.

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Retirado em Cosine similarity

- **Clusterização baseada em centróide**

A metodologia mais conhecida e que engloba o algoritmo K-means citado neste trabalho. Clusters são formados a partir de proximidade, calculado por alguma métrica, com um determinado centróide (que não necessariamente pertence ao conjunto de dados). No caso do K-means, são realizadas diversas iterações com objetivo de reposicionar os centróides. Para este tipo de abordagem, é necessário que o número de clusters seja especificado previamente.



Retirado em [Clustering Algorithms | Clustering in Machine Learning | Google Developers]

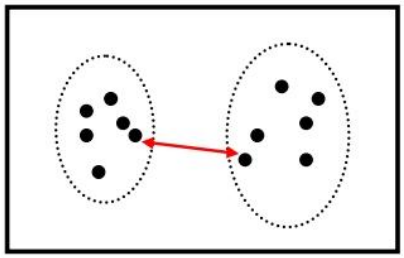
- **Clusterização hierárquica**

Realiza a clusterização a partir de diversos clusters que são mesclados ao longo da execução. Existem duas abordagens presentes neste tipo de algoritmo:

- Método aglomerativo: inicialmente cada ponto de dado é um cluster e posteriormente passa-se a agrupar com outros clusters mais próximos. Essa abordagem é também conhecida como bottom-up.

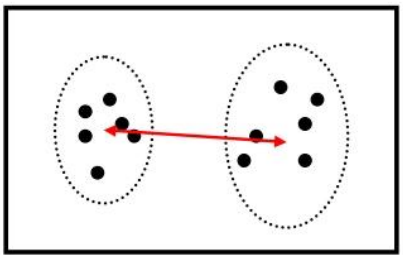
- Método divisível: oposto ao aglomerativo (top-down). No início todos os pontos de dados englobam apenas um cluster que vai se separando em diversos clusters ao longo da execução.

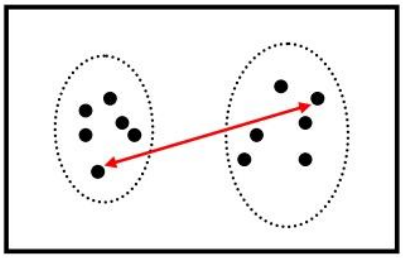
Para realizar estas junções (no caso do aglomerativo) e separações (no caso do divisível) é necessário um critério de linkagem. Abaixo são apresentados três critérios bastante utilizados:



Single Link: Distância entre dois clusters é a distância entre os pontos mais próximos. Também chamado “agrupamento de vizinhos”

Average Link: Distância entre clusters é a distância entre os centróides

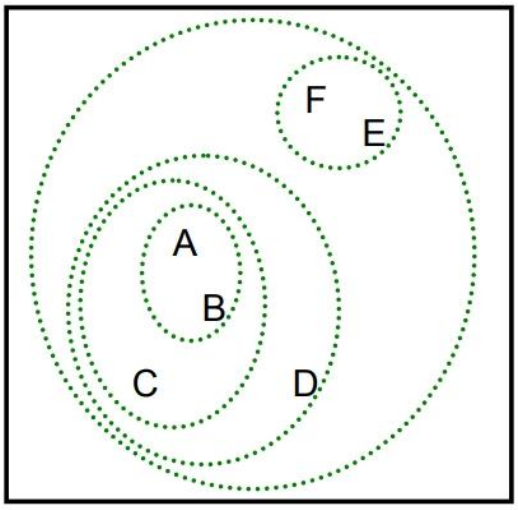


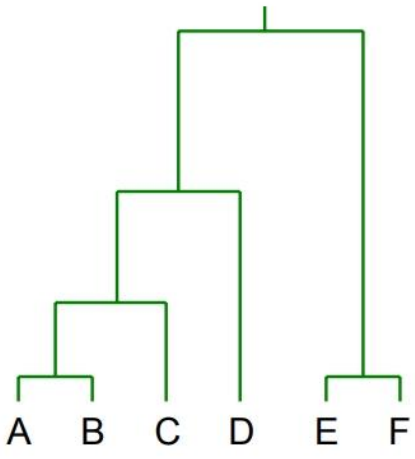


Complete Link: Distância entre clusters é a distância entre os pontos mais distantes

Retirado em [Clustering (Agrupamento)]

Este tipo de clusterização tem como característica gerar um dendrograma, por isso a natureza do nome "hierárquico". Para definir a quantidade de clusters, basta realizar um corte horizontal (eixo-y) neste diagrama. Veja abaixo:



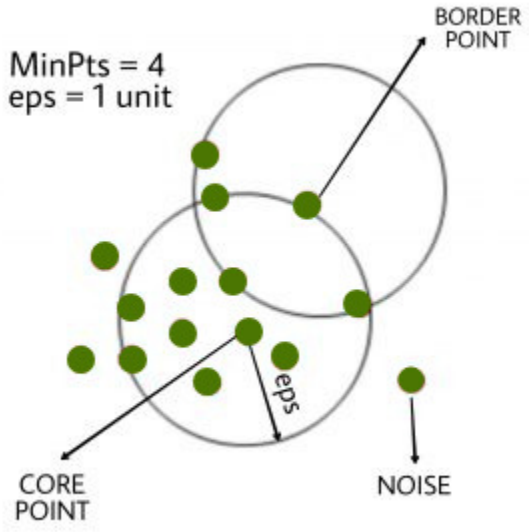


Retirado em [Clustering (Agrupamento)]

- **Clusterização baseada em densidade**

Agrupa pontos de dados baseado em regiões na qual a quantidade de objetos de dados excede um determinado valor de referência. A maioria dos algoritmos que utilizam esta técnica requer dois parâmetros:

- ε: define o tamanho da vizinhança em torno de um ponto. Ou seja, o tamanho ou raio do cluster.
- MinPts: número mínimo de vizinhos (pontos de dados) dentro do raio de ε.



Retirado em [DBSCAN Clustering in ML | Density based clustering]

K-means

Como citado anteriormente, o algoritmo K-means preenche o grupo de algoritmos para clusterização baseados em centróides. Nesta seção será abordado mais aprofundadamente este método.

- **Abordagem**

O algoritmo K-means busca clusterizar os pontos de dados por meio da minimização do valor de SSE (sum-of-squared-errors). Para um conjunto de pontos no espaço euclidiano, encontrar um conjunto C de k pontos tal que a soma do quadrado das distâncias dos pontos até o centro C mais próximo é mínima. Sendo assim, a função objetivo é:

$$\operatorname{cost}(P, C) := \sum_{p \in P} \min_{c \in C} \|p - c\|^2$$

Retirado em Theoretical Analysis of the k-Means Algorithm - A Survey

No k-means, primeiramente é definido o número de clusters a serem formados (K), além do local onde serão localizados os centros iniciais. Existem diversas maneiras para definir estes parâmetros, as quais são discutidas posteriormente neste trabalho. Após isso, é realizado um processo iterativo até que não exista mais mudança na localização dos centróides ou que seja atingido um número de iterações pré-definido.

Quando não há mais mudanças na localização dos centróides diz-se que o algoritmo convergiu.

Os centróides correspondem a média aritmética dos pontos pertencentes ao respectivo cluster e cada ponto está mais próximo do centro do seu cluster do que de qualquer outro centro de cluster.

A maneira como o K-means atinge a solução é a seguinte:

- Defina alguns centros
- Repita até convergir ou atingir número máximo de iterações
    - E-Step: atualize pontos aos seus clusters mais próximos (baseado em métrica de distância).
    - M-Step: atualiza os centróides com base na média

Abaixo é apresentado seu pseudo-código, retirado em K-means:

K-MEANS($\{\vec{x}_1, \ldots, \vec{x}_N\}, K$)
1 $(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS($\{\vec{x}_1, \ldots, \vec{x}_N\}, K$)
2 **for** $k \leftarrow 1$ **to** K
3 **do** $\vec{\mu}_k \leftarrow \vec{s}_k$
4 **while** stopping criterion has not been met
5 **do for** $k \leftarrow 1$ **to** K
6 **do** $\omega_k \leftarrow \{\}$
7 **for** $n \leftarrow 1$ **to** N
8 **do** $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
9 $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
10 **for** $k \leftarrow 1$ **to** K
11 **do** $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
12 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

No final deste trabalho é apresentado tanto o código de uma implementação manual do K-means, assim como uma aplicação que utiliza a implementação disponível na popular biblioteca scikit-learn (ambos em Python).

- **Complexidade de tempo**

Considerando d como o tempo para cálculo de distância, n o número de pontos de dados e k como o número de clusters, a complexidade de tempo do K-means pode ser escrita, para apenas uma iteração, como:

$$\Theta\left(ndk\right)$$

- **Aplicações**

A quantidade de aplicações que utilizam clusterização e algoritmos de clusterização é infinita. Abaixo são citados alguns destes exemplos:

- **Compressão de cores:**

Uma aplicação interessante do algoritmo K-means é a compressão de cores em imagens. Dentro de uma imagem existem muitos pixels (dependendo da resolução). No entanto, muitos pixels são bastante similares e a visualização destes passa a ser dificilmente distinguível. Sendo assim, é possível separar estes pixels bastante semelhantes e produzir uma imagem eficiente utilizando menos bytes:

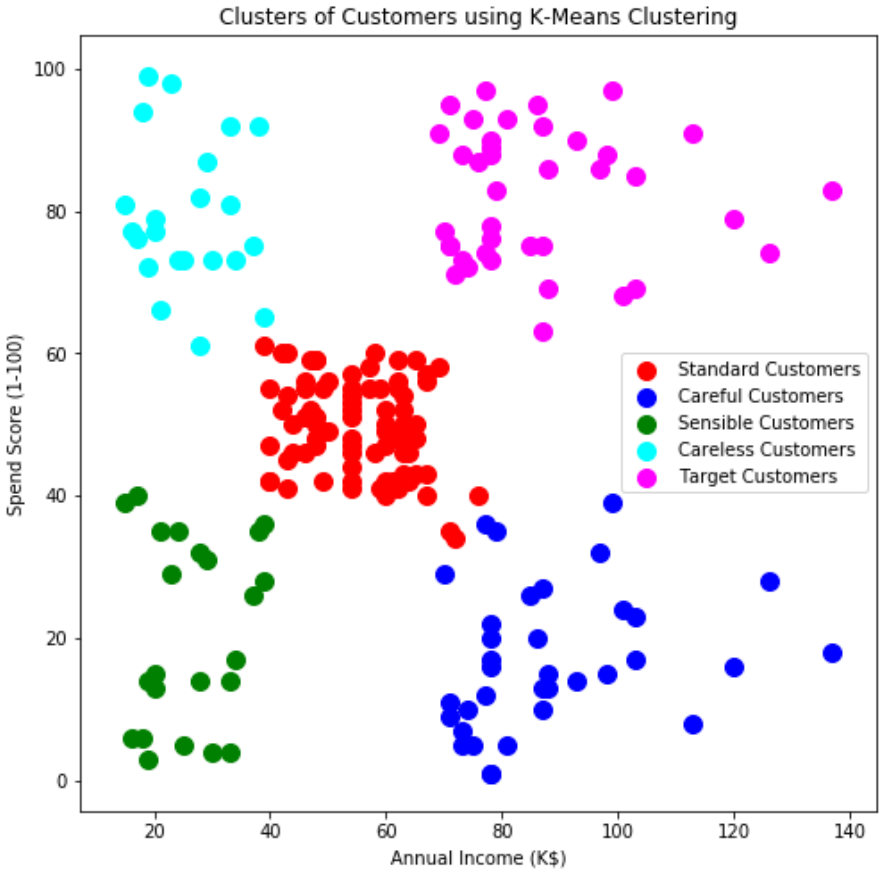
Retirado em Image Compression using K-Means Clustering | by Satyam Kumar | Towards Data Science

- **Agrupamento de documentos:**

É possível separar grupos de documentos baseado nos seus tópicos, tags e o conteúdo dos mesmos utilizando o algoritmo k-means. Pelo fato do k-means necessitar de dados numéricos, utiliza-se alguma técnica para cálculo de features, como frequência de termos, score TF-IDF, etc.

- **Segmentação de consumidores:**

Segmentação de consumidores corresponde à divisão em grupos de clientes que compartilham características similares. Com base nestes grupos é possível determinar ações relacionadas a estes resultados, como identificar consumidores insatisfeitos, descobrir tendências e interesses, aplicar bonificações, etc. Veja um exemplo abaixo:



Retirado em [Compare K-Means & Hierarchical Clustering In Customer Segmentation]

Nesta imagem, os grupos são constituídos baseados em features de poder aquisitivo e gasto realizado. Por exemplo, clientes com baixo poder aquisitivo e alta quantidade de gastos são separados no grupo “Clientes descuidados”.

Desvantagens do K-means

Apesar do k-means ser bastante útil, como todo algoritmo, ele possui suas desvantagens. Abaixo são apresentadas algumas delas:

- **Dependência de centróides iniciais**

O local onde os centróides são iniciados possuem um impacto significante nos resultados obtidos. Este panorama é melhor discutido na seção seguinte, sobre métodos de inicialização.

A figura abaixo mostra a diferença entre duas execuções diferentes. Esse exemplo demonstra como a posição de início dos centróides é importante para determinar a melhor representação dos clusters. Veja como a figura 1.2 apresenta melhores resultados comparado a figura 1.1 (isto porque a figura 1.2 possui melhores valores da função objetivo):

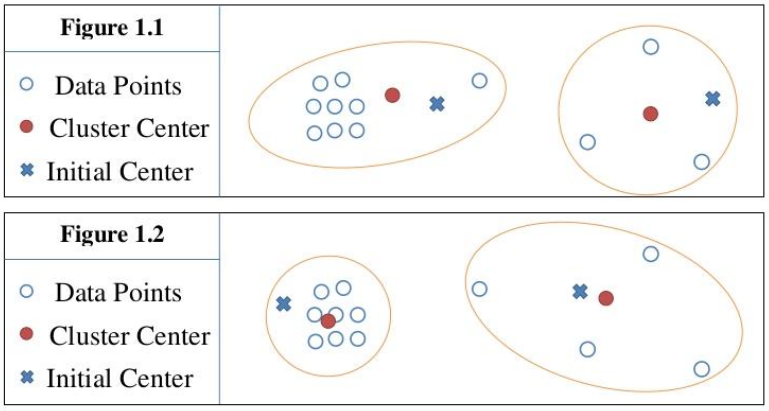


Fig. 1: Initial centroid effects on K-means result.

Retirado em [DIMK-means "Distance-based Initialization Method for K-means Clustering Algorithm"]

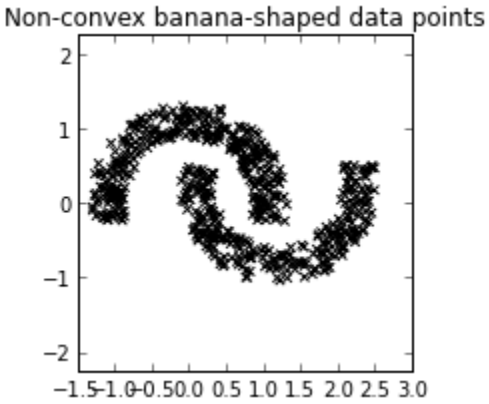
- **Necessidade de especificar K**

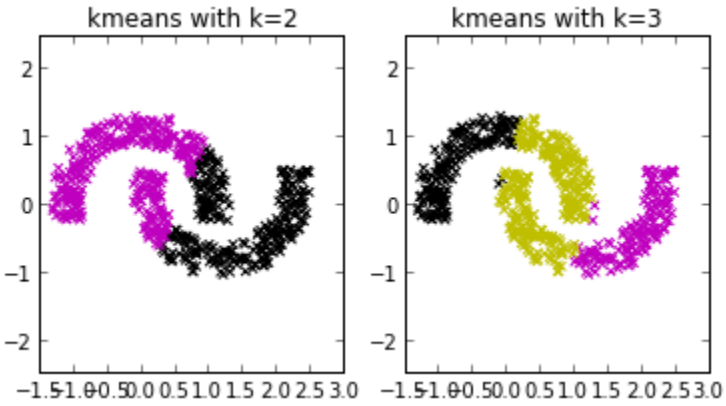
O valor de K é um parâmetro necessário para a execução do K-means e afeta diretamente o desempenho do algoritmo. No entanto, tal abordagem não é capaz de aprender o número de clusters a partir dos dados. Como consequência disto, existem diversas heurísticas que buscam encontrar o valor ideal.

Determinar o valor de K automaticamente vem sendo um dos assuntos mais complicados em clusterização. Na maioria dos casos, o algoritmo é executado diversas vezes com valores de K diferentes e o melhor resultado é escolhido baseado em um critério previamente estabelecido. Alguns desses métodos são abordados e aprofundados na seção "Escolha de K".

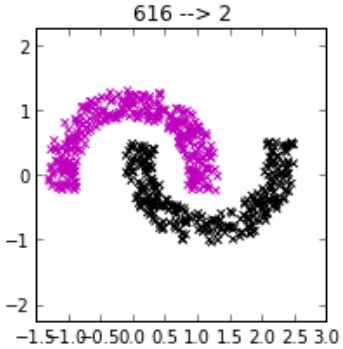
- **Conjunto de dados não-convexo**

Por natureza o K-means tem como tendência formar clusters esféricos ao redor de um centróide. Portanto, para conjunto de dados não-convexos, o agrupamento gerado pelo algoritmo pode não ser o ideal. Veja abaixo um exemplo deste comportamento, retirado em [Non-convex sets with k-means and hierarchical clustering | pafnuty.blog]:





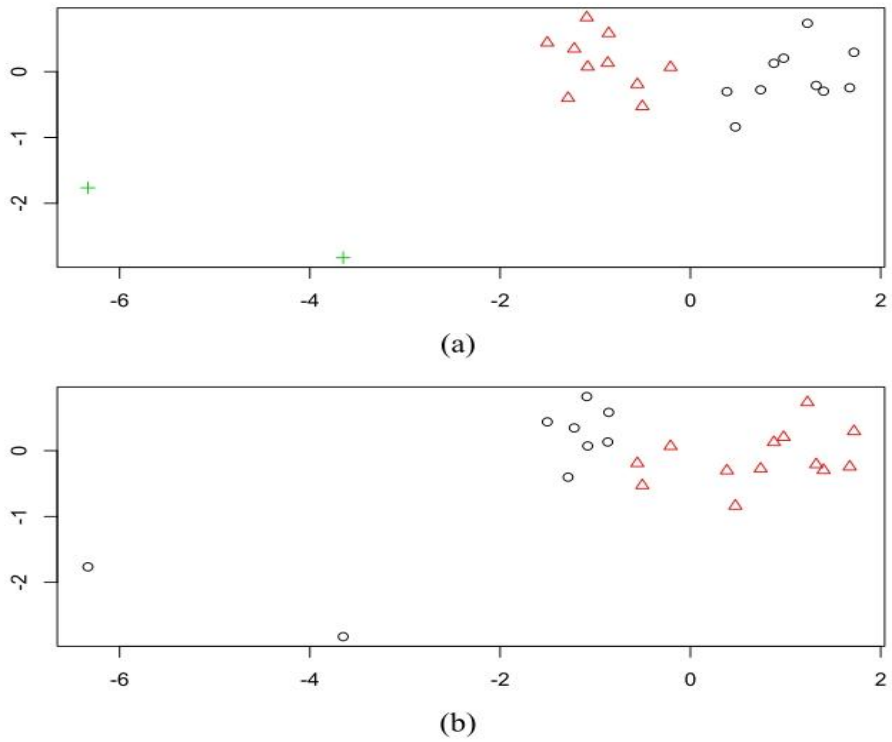
No entanto, outras formas de clusterização, como baseada em hierarquia, são capazes de performar melhor neste tipo de cenário:



- **Outliers**

Outliers são dados que possuem valores muito distintos do conjunto de dados que compõem, destoando dos demais. Muitas vezes são dados gerados incorretamente durante a etapa de construção do dataset.

Durante o processo de agrupamento dos dados, o k-means utiliza diversas vezes o cálculo de média (que é influenciado por extremos) para centróides, buscando clusterizar todos os pontos de dados possíveis. Devido a isto, o resultado produzido pode estar distorcido em decorrência dos outliers presentes. Veja abaixo um exemplo deste comportamento:



Retirado em k -means clustering with outlier removal

Uma solução para este problema seria submeter o dataset a um pré-processamento, com o intuito de remover estes outliers.

Métodos de inicialização

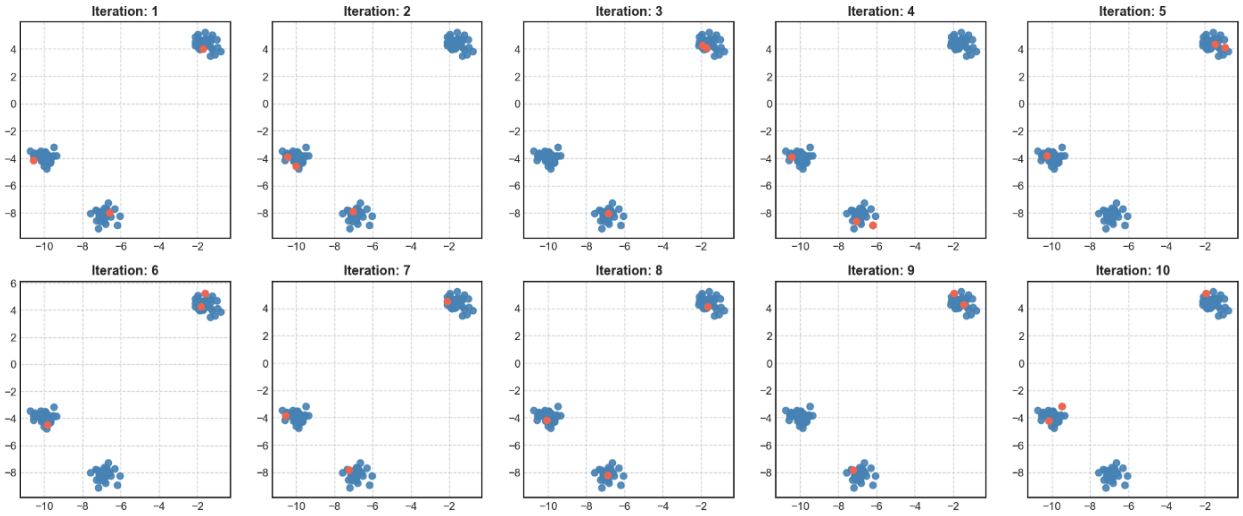
Como visto anteriormente, o algoritmo K-means para clusterização sofre de efeitos diretos proporcionados por condições iniciais. Dependendo da forma como os clusters são iniciados na primeira iteração, o desempenho pode ser diferente a cada execução, influenciando o resultado final observado. Sendo assim, é necessário conhecer algumas das abordagens utilizadas para iniciar clusters.

Serão apresentados 3 métodos bastante populares: partição randômica, forgy e kmeans++. Todos estes pertencem ao grupo de métodos que requerem complexidade linear para serem executados. Posteriormente é apresentado a comparação dos resultados empíricos obtidos pelos MI disponibilizados em [1].

- **Inicialização Forgy**

O método de inicialização Forgy, primeiramente introduzido por Edward Forgy, é bastante simples e rápido. Nele, são escolhidos K pontos aleatórios de dados do dataset. O benefício deste método é claro: escolher pontos de dados como centroide inicial geralmente garante proximidade com os outros dados. Em contrapartida, em alguns casos, a escolha randômica de centróides pode causar grande demora para convergir ou então convergir para uma configuração ruim.

Abaixo é apresentado 10 inicializações diferentes utilizando Forgy:



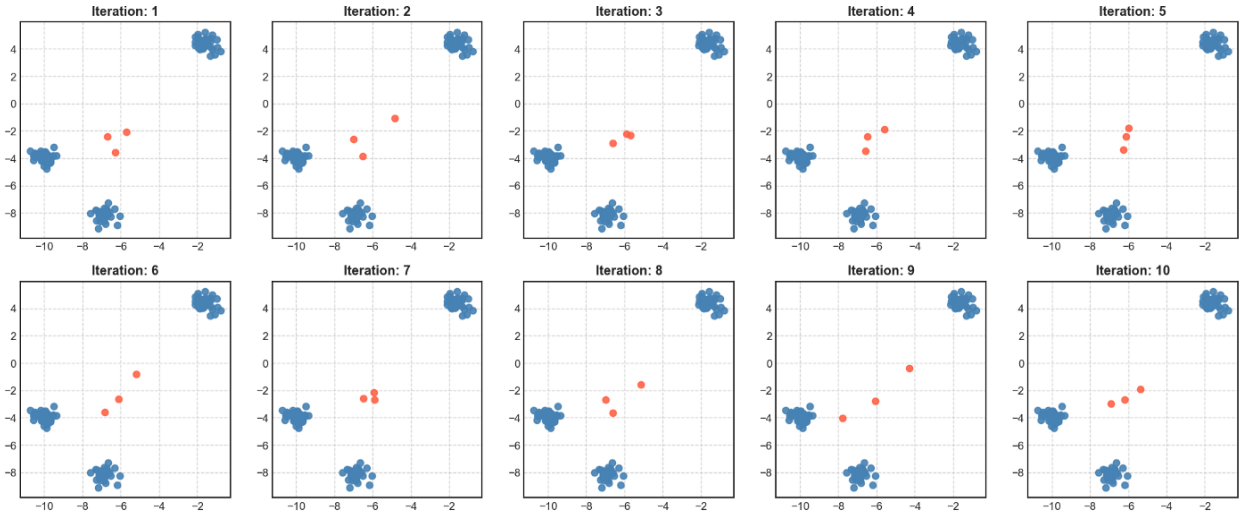
Retirado em [kMeans: Initialization Strategies- kmeans++, Forgy, Random Partition | Analytics Vidhya]

É possível observar que as inicializações 1, 7 e 8 possuem uma boa distribuição de clusters dentro do conjunto de dados. No entanto, o mesmo não acontece para as demais

inicializações, sendo encontrados centróides muito próximos. Nestes casos, existe uma grande possibilidade de que os resultados finais sofreram consequências negativas destas escolhas de começo.

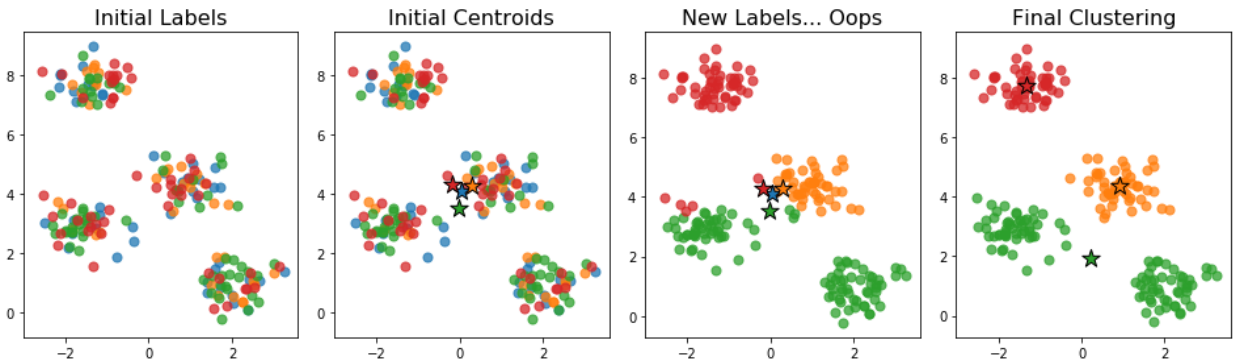
- **Inicialização por partição randômica**

Neste caso, inicialmente os pontos de dados são associados aleatoriamente a grupos de clusters (partições). Posteriormente, obtém-se a média dos dados de cada partição formada e então tem-se os valores de centróides iniciais. Este método tem como tendência localizar os centróides próximos ao centro global dos dados.



Retirado em [kMeans: Initialization Strategies- kmeans++, Forgy, Random Partition | Analytics Vidhya]

Como ponto negativo, se houver um centro inatingível, a quantidade de clusters tende a cair. Veja por exemplo a estrela azul no exemplo abaixo:



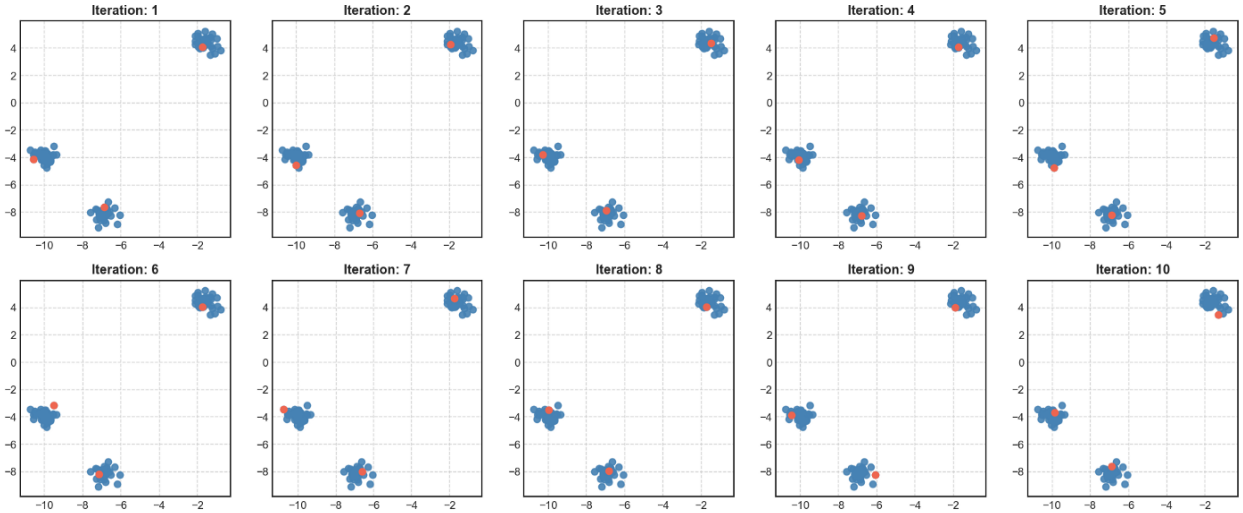
Retirado em [K-Means Clustering Part 2]

- **K-means++**

O método de inicialização K-means++ segue um pouco da ideia apresentada na abordagem de Forgy. No entanto, K-means++ prioriza a seleção de pontos de dados mais distantes dos centróides selecionados, ou seja, uma distribuição mais espalhada dos centros de clusters. Toda essa abordagem é feita baseada na distribuição de probabilidade dos dados em relação aos centróides mais próximos.

Seja D(x) o valor que denota a menor distância de um ponto de dado ao seu centróide mais próximo já escolhido e X o conjunto de dados, tem-se o seguinte passo-a-passo:

1a. Tome um centro c1 escolhido aleatoriamente de X
1b. Tome um novo centro ci, escolhido a partir de X com probabilidade $\frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2}$
1c. Repita 1b até ter K centros obtidos

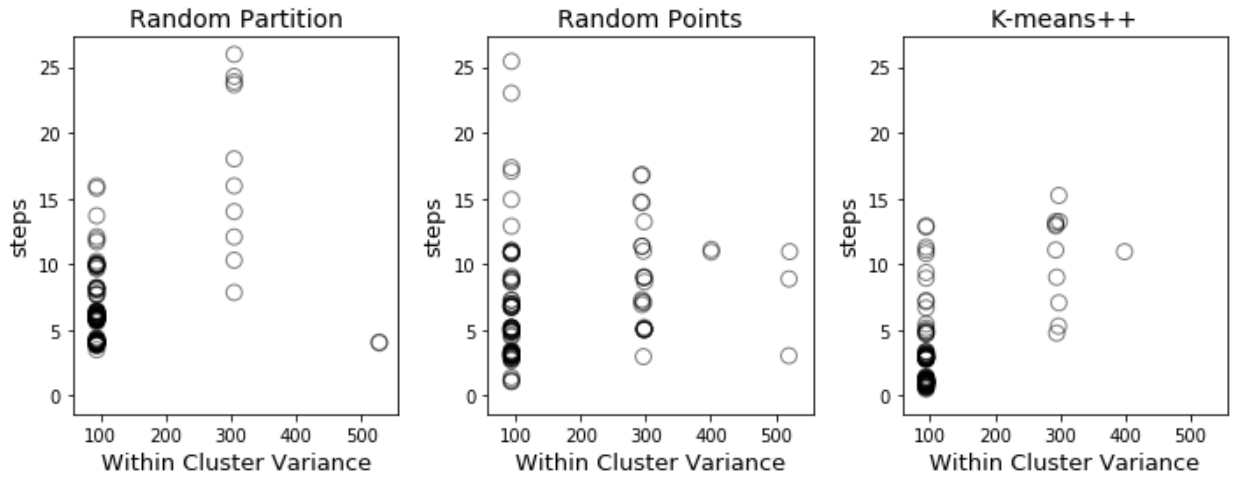


Retirado em [kMeans: Initialization Strategies- kmeans++, Forgy, Random Partition | Analytics Vidhya]

Este é o método mais recomendado, encontrado como padrão nas mais populares implementações do K-means, como na biblioteca scikit-learn.

- **Comparação entre os 3 MI**

Com base no experimento demonstrado em K-Means Clustering Part 2, onde foi executado 100 diferentes inicializações para cada método, o autor obteve os seguintes resultados:



Pode-se observar que o método de inicialização K-means++, como esperado, possui a menor quantidade de passos para ser executado (ou seja, converge mais rapidamente) devido a sua abordagem de distribuição inicial de centróides. Além disso, independente do método, todos obtiveram o valor mínimo de Within Cluster Variance, sugerindo a necessidade de execução repetida para qualquer panorama de MI escolhido.

Escolha de K

A escolha do número de clusters a serem formados está diretamente ligado ao desempenho dos algoritmos de clusterização. Neste caso, para o K-means, existem diversos métodos que auxiliam na busca de um valor de K que proporcione melhor eficiência no agrupamento dos dados. Neste documento serão apresentados duas abordagens bastante populares, o Método do Cotovelo (Elbow Method) e o Método da Silhueta (Silhouette Method):

- **Método do Cotovelo**

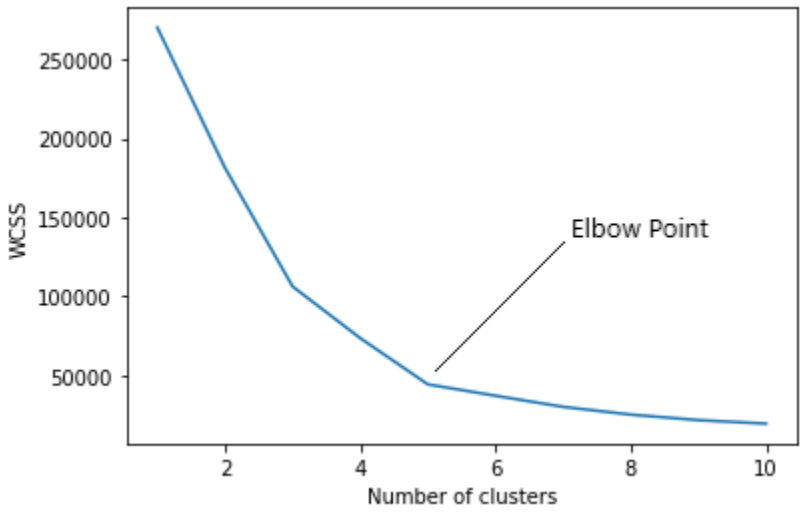
Este método utiliza o parâmetro WCSS (*Within Cluster Sums of Squares*) ou SSE(Sum-of-squared-errors) como referência. O valor de WCSS é estimado pela soma do quadrado da distância de cada ponto em relação ao seu respectivo cluster, ou seja, a soma das distâncias euclidianas entre os centróides e os pontos de dados disponíveis, ou se preferir, a soma dos valores de SSE para cada dado:

C_i = Cluster, N_c = # clusters, $\overline{x}_{c_i}$ = Cluster centroid,

- **Within Cluster Sums of Squares :** $$\mathrm{WSS} = \sum_{i=1}^{N_C} \sum_{x \in C_i} d(\mathbf{x}, \bar{\mathbf{x}}_{\mathbf{C_i}})^2$$

Retirado em [Who is your Golden Goose?: Cohort Analysis - KDnuggets]

Naturalmente, com o aumento do número de clusters, o valor de WCSS tende a 0. Isto é, quando o número de K é o mesmo que o comprimento do dataset. Sendo assim, este método está interessado em selecionar o valor de K específico onde a variação de WCSS passa a ser pequena, ou seja, quando a mudança de K não impacta significativamente o modelo. Veja abaixo um exemplo de gráfico WCSS versus K para um dataset aleatório:



Retirado em [In-depth Intuition of K-Means Clustering Algorithm in Machine Learning]

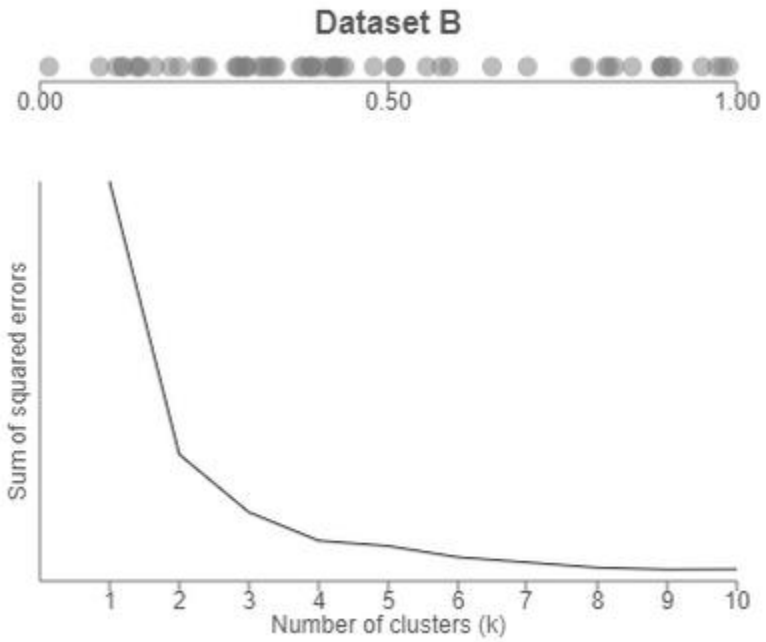
É possível observar que a partir de (K = 5) existe uma redução na variação brusca do valor de WCSS, sendo assim, formando uma espécie de cotovelo (origem do nome). Portanto, representando o melhor número de clusters a serem formados.

Além disso, é válido comentar que este ponto K, onde ocorre a presença deste “cotovelo”, está localizado graficamente no ponto de maior distância em relação a uma reta que liga as extremidades do gráfico. Sendo assim, o valor de K ótimo pode ser encontrado pelo maior valor encontrado na fórmula matemática abaixo quando executada para todos os valores de K:

$$\text{distance}(P_0, P_1, (x, y)) = \frac{|(y_1 - y_0)x - (x_1 - x_0)y + x_1 y_0 - y_1 x_0|}{\sqrt{(y_1 - y_0)^2 + (x_1 - x_0)^2}}$$

Retirado em [Como definir o número de clusters para o seu KMeans | by Jessica Temporal | pizzadedados | Medium]

No entanto, nem sempre o método do cotovelo pode ser utilizado para encontrar o valor de K. Observe o exemplo abaixo de do gráfico (WCSSxK) para um dataset aleatório:

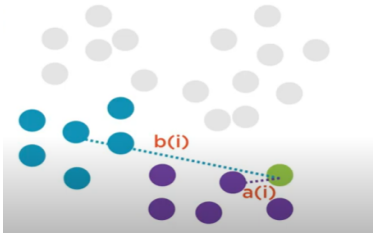


Retirado em [How to Determine the Optimal K for K-Means? | by Khyati Mahendru | Analytics Vidhya | Medium]

Neste caso, existe uma ambiguidade presente no gráfico que não possibilita distinguir o ponto específico da variação. Sendo assim, decorrente deste comportamento, o método da silhueta pode ser utilizado para solucionar este tipo de ocasião. A seguir é apresentado esta metodologia.

- **Método da Silhueta**

O método da silhueta representa outra metodologia disponível para encontrar valores ótimos para a quantidade de clusters. A abordagem da silhueta busca descobrir o quanto similar um determinado dado é em relação ao seu cluster (coesão) comparado a outros clusters (separação) a partir de um valor, chamado silhueta (s), variando de [-1, 1]. Com isso, podemos ter as seguintes conclusões para alguns valores de s associados a K:



Retirado em [Silhouette Method — Better than Elbow Method to find Optimal Clusters | by Satyam Kumar | Towards Data Science]

+1: Clusters são perfeitamente distinguíveis
0: Clusters são neutros e não distinguíveis
-1: Clusters estão organizados erroneamente

Portanto, quanto mais perto do valor numérico 1, melhor será a escolha para o valor de K. Sendo assim, estamos interessados no maior valor numérico de silhueta encontrado.

O método da silhueta utiliza a seguinte fórmula para medir o quão bem um ponto de dado está associado ao seu respectivo cluster:

Para cada ponto de dado $i \in C_I$ *(dado i no cluster CI),*

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i,j)$$

Retirado em [Silhouette (clustering) - Wikipedia]

Onde a(i) representa a distância média entre o ponto e os seus vizinhos pertencentes a seu cluster. Em vista disso, d(i, j) pode ser encontrado utilizando qualquer métrica de distância, como por exemplo euclidiana, | CI | equivale ao número de pontos no cluster e quanto menor o valor de a(i), melhor foi o agrupamento encontrado pelo algoritmo de clusterização (neste caso, k-mean).

Além disso, é necessário encontrar a dissimilaridade do ponto em relação aos outros clusters. Portanto, utiliza-se o resultado mínimo entre a média da distância de i para todos os pontos em Cj, onde Cj equivale a um outro cluster disponível:

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$$

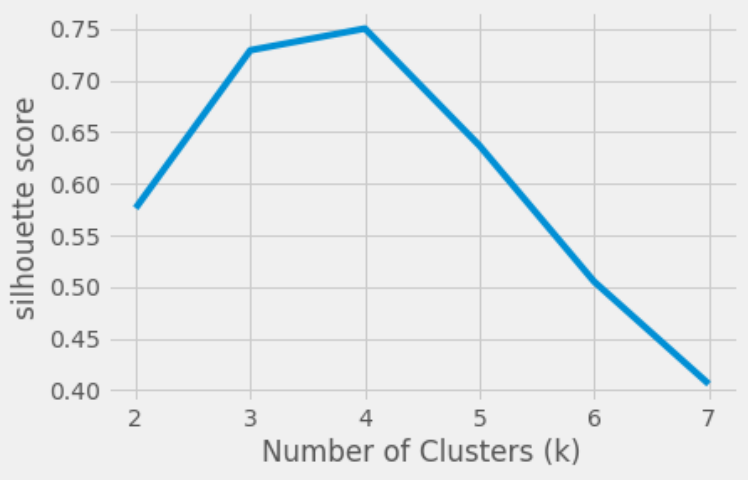
Retirado em [Silhouette (clustering) - Wikipedia]

Com este resultado é possível encontrar o cluster mais próximo do cluster de um determinado ponto. Partindo destes valores em mãos, basta calcular o valor s(i) de silhueta:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Retirado em [Silhouette (clustering) - Wikipedia]

Como exemplo para concretizar a ideia, observe o seguinte gráfico (Score Silhouette x K):



Retirado em [Silhouette Method — Better than Elbow Method to find Optimal Clusters | by Satyam Kumar | Towards Data Science]

Neste caso, o valor de K ótimo seria (K = 4), onde ocorre o pico dos scores de silhueta encontrados.

Implementação K-means

https://colab.research.google.com/drive/1U_Bwv0aq54IuStvnHWZkh1OjhjTl5Xpi?usp=sharing

Aplicação

https://colab.research.google.com/drive/1ZBfJ_iRbMEEODlvDSRLZm1N2DjL0Hsay?usp=sharing

Bibliografia

Data Mining na Prática: Algoritmo K-Means (devmedia.com.br)
K-means: o que é, como funciona, aplicações e exemplo em Python | by Bruno Anastacio | Programadores Ajudando Programadores | Medium
k-means clustering - Wikipedia
Visualizing K-Means algorithm with D3.js - TECH-NI Blog
Como definir o número de clusters para o seu KMeans | by Jessica Temporal | pizzadedados | Medium
Silhouette (clustering) - Wikipedia
https://arxiv.org/pdf/1209.1960.pdf
http://www.salientiastuff.com/k-means-clustering-part-2.html
http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf
https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html
What is Unsupervised Learning? | IBM

Resumo k-means apresentação
Aprendizado não-supervisionado
- Abordagem
- Exemplo

Clusters
- Abordagem
- Exemplo

K-mean
- **Abordagem**
- **Detalhes e fórmulas**
    - **SSE**
- **Métodos de inicialização**
    - **Random Partition**
    - **Kmeans++**
    - **Forgy**
- **Escolha de K**
    - **Elbow Method**
    - **Silhouette Method**
- **Desvantagens do K-means (spherical forms, outliers, etc)**
- **Aplicações**
- **Implementação (código)**
- **Exemplo com dataset/scikit-learn/matplotlib**